Final Report

Project Title: Amorphous and Cellular Computing
Organization: Massachusetts Institute of Technology (MIT)
AO Number: E569
Contract Number: N00014-96-1-1228
Start Date: 01 Sept. 1996
End Date: 31 Aug. 2001

Principal Investigators

Harold Abelson
MIT Artificial Intelligence Laboratory Cambridge, MA 02139
617-253-5856
hal@zurich.ai.mit.edu

Gerald Jay Sussman
545 Technology Square
Cambridge, MA 02139
617-253-5874
gjs@zurich.ai.mit.edu

Thomas F. Knight, Jr.
MIT Artificial Intelligence Laboratory
Cambridge, MA 02139
tk@mit.edu

Scientific and Technical Objectives

The objective of this research is to create the architectural, algorithmic, and technological foundations for exploiting programmable materials. These are materials that incorporate vast numbers of programmable elements that react to each other and to their environment. Such materials can be fabricated economically, provided that the computing elements are amassed in bulk without arranging for precision interconnect and testing. In order to exploit programmable materials we must identify engineering principles for organizing and instructing myriad programmable entities to cooperate to robustly achieve pre-established goals, even though the individual entities are unreliable and interconnected in unknown, irregular, and time-varying ways.

Progress in microfabrication and in bioengineering will make it possible to assemble such *amorphous systems* at almost no cost, provided that 1) the units need not all work correctly; 2) the units are identically programmed; and 3) there is no need to manufacture precise geometrical arrangements of the units or precise interconnections among them.

Harnessing these systems for information processing and intelligent materials requires augmenting traditional information processing with new perspectives and techniques from biology.

Computer science is currently built on a foundation that largely assumes the existence of a perfect infrastructure. Integrated circuits are fabricated in clean-room environments, tested deterministically, and discarded if even a single defect is uncovered. Entire software systems fail with single-line errors. In contrast, biological systems rely on local computation, local communication, and local state, yet they exhibit tremendous resiliance. No existing engineering framework creates such complex structure from simple, potentially faulty components or maintains such complex behavior of existing structures under dramatic environmental change.

Our new framework combines inspiration about robust design from biology—from morphogenesis and development—with the techniques of organization and control of complexity from computer science.

An effective integration of these perspectives will have a profound impact on computer science, biology, and microelectronics. In biology, we can perhaps address long-standing questions of morphogenesis, and discover a computational understanding of the developmental process. In computer science, we can engineer robust computational infrastructures, out of unreliable components. In microelectronics, we can biochemically pattern nanoscale information rich substrates with atomic precision. We can incorporate robust local computation, sensing, and effectors arrays into structures which exhibit sophisticated global behavior..

Our new framework helps provide the power to take control of some biological processes, and to design and construct biological cells and cell assemblies with prespecified behaviors. Such controlled biological mechanisms will give us the ability to construct of novel materials with engineered nanoscale structures.

Approach

In order to exploit programmable materials we must identify engineering principles for organizing and instructing myriad programmable entities to cooperate to robustly achieve pre-established goals, even though the individual entities are unreliable and interconnected in unknown, irregular, and time-varying ways.

We demonstrate the feasibility of these principles in prototype amorphous systems, implemented both in traditional silicon technology and in novel computational substrates that exploit molecular biology. This includes two major thrusts:

1. Invent new programming paradigms and languages for controlling amorphous computing agents.

    - We extend the simulation technology we have already developed to aid in the design and testing of algorithms for amorphous computers.

- We have already demonstrated that amorphous media can be configured by a program, common to all computing elements, to generate highly complex prespecified patterns. For example, we can specify that an amorphous medium manifest a pattern representing the interconnection structure of an arbitrary electrical circuit.
- We further develop high-level languages for expressing algorithms for amorphous computers. They will provide support for descriptions that do not depend on precision interconnect or perfectly working parts. The concepts involved in the description of an algorithm will be conserved in the face of slow deformation or flow of the amorphous elements.
- We develop languages and programming paradigms for amorphous systems in which the individual elements are sensitive to their physical environment and can affect it by changing their shapes. If appropriately programmed, collections of such elements can provide an implementation basis for self-configuring physical structures.
- We develop specific applications to demonstrate the utility of the amorphous computer. These applications include methods for the solution of partial differential equations, and coordination and processing of data collected from sensors distributed throughout an amorphous computer.

2. Investigate prototypes, both in traditional silicon technology and in molecular biology.

   - We have demonstrated, by simulation of reaction kinetics, the theoretical feasibility of building a family of logic gates where the signals are represented by concentrations of DNA-binding proteins, and where the nonlinear amplification is implemented by in vivo DNA-directed protein synthesis.
   - We have installed and equipped a complete recombinant DNA laboratory for constructing and characterizing "cellular logic" gates, based on these theoretical foundations, and we are using this laboratory to construct novel organisms incorporating small digital circuits.

Highlighted summary accomplishments

- *Approaches to programming amorphous systems:*

  We have learned that it is possible to engineer prespecified, coherent behavior from the cooperation of immense numbers of unreliable parts that are interconnected in unknown, irregular, and time-varying ways. We have found that biology can be a powerful source of organizational principles for this kind of engineering. We have developed novel programming models that encapsulate these organizational principles. For each of these models we have found general methods of compiling

programs that describe global behavior into instructions for the components in such a way that all of the components are identically programs and the behavior of the aggregate is robust in the face of unreliable components or communications.

- One model allows us to control global topology. Here programs are organized according to a botanical metaphor where "growing points" and "tropisms" control the differentiation of amorphous computing agents to form the various elements of the pattern.
- Another model allows us to control global geometry. We allow the components to change shape individually. The programs that specify the target geometry are organized as folding sequences that could describe an origami construction.

We have developed a prototype model for controlling programmable materials. We have shown how program sheets consisting of identically programmed and irregularly placed agents, that can individually deform to cooperate to construct a large family of globally-specified shapes.

- *Programmable cells as a computing substrate:*

  We have learned how to instruct living cells to execute some of the behaviors that are required to act as the components of a programmable amorphous medium. Biological cells are self-reproducing chemical factories that are controlled by a program written in the genetic code. As engineers, we can take control of this process to make novel organisms with particular desired properties. We have found that it is feasible to build a family of logic gates where the signals are represented by concentrations of naturally-occurring DNA-binding proteins, and where the nonlinear amplification is implemented by *in vivo* DNA-directed protein synthesis.

  We have demonstrated that one can engineer intercellular communication in bacterial colonies, through the techniques of cellular logic. We isolated the gene clusters responsible for light production, autoinducer production, and autoinducer response in *Vibrio fischeri.* The gene clusters were split to make a transmitter and a receiver and inserted into *E. coli.* Stimulating the transmitter colony releases a chemical signal that causes the receiver colony to light up.

  We have demonstrated the fundamentals of "genetic process engineering" – taking existing genertic regulatory elements and modifying their DNA encoding so that they can be used in constructing complex *in vivo* digital-logic circuits.

Selected accomplishments

The Growing Point language Daniel Coore's 1999 Ph.D. thesis demonstrates that amorphous media can be configured by a program, that is common to all the computing

elements, to generate highly complex pre-specified patterns, for example, the pattern representing the interconnection structure of an arbitrary prespecified electrical circuit.

Coore's strategy is inspired by a botanical metaphor based on growing points and tropisms. To make this strategy explicit, he developed the Growing Point Language (GPL). A growing point is a locus of activity in an amorphous medium. A growing point propagates through the medium by transferring its activity from one computing element to a neighbor. As a growing point passes through the medium it effects the differentiation of the behaviors of the computing elements it visits. The trajectory of the growing point is controlled by signals that are automatically carried through the medium from other differentiated elements. Such a response is called a tropism. In this way a GPL program can exploit locality to make crude geometric inferences.

There is a wide variety of patterns that are expressible in GPL. Examples include: Euclidean constructions, branching structures and simple text. Coore proves that amorphous media can be programmed to draw any prespecified planar graph, and obtains upper bounds on the amount of storage required by the individual processors to realize such a graph. He also analyzes how the effectiveness of GPL programs depends upon the distribution of the computing elements.

Marker propagation for amorphous particles GPL is formulated in terms of abstractions that must ultimately be implemented by processes in the individual computational particles, which we assume are all programmed identically. Ron Weiss (1999) developed a remarkably convenient and simple language for programming the particles. In this model, the program to be executed by each particle is constructed as a set of independent rules. The state of each particle includes a set of binary markers, and rules are enabled by boolean combinations of the markers. The rules that are enabled are triggered by the receipt of labelled messages from neighboring particles. A rule may set or clear various markers, and it may send further messages. A message is count that determines how far it will diffuse, and a marker has a lifetime that determines how long its value lasts. Underlying this model is a runtime system that automatically propagates messages and manages the lifetimes of markers, so that the programmer need not deal with these operations explicitly. Weiss's system is almost powerful enough to represent the processes described by Coore's growing points, yet it is simple enough that it can be implemented in an elementary way. It does not depend on any arithmetic or data structures, and it would be an obvious candidate for implementation by real biological cells and the cellular computing.

Algorithms for self-organizing communication networks Radhika Nagpal (unpublished, 1997) developed new algorithms for self organising communication networks for large distributions of smart sensors and actuators. A network of routes and routers emerges as a result of the communication pattern. The source finds the the fastest route to a destination destination by using a search gradient that is sensitive to congestion. Routes are maintained depending on the frequency with which they are used and are self

repairing. The set of dominant (long-term) routes tends to mimic the dominant communication pattern and thus automatically forms an efficient network for the given problem. This work was done in the Measurements Systems Lab at Hewlett Packard Laboratories, Palo Alto.

Cellular-logic circuit design Gerald Jay Sussman and Tom Knight (1998) demonstrated, by simulation of reaction kinetics, the theoretical feasibility of building a family of logic gates where the signals are represented by concentrations of DNA-binding proteins, and where the nonlinear amplification is implemented by in vivo DNA-directed protein synthesis. They determined that the nonlinearities required to make devices with excellent noise margins, that are robust to large variations in the kinetic constants, can be achieved using tetramer binding proteins.

This work showed the theoretical possibility of building cellular logic in terms of single-input inverters. Ron Weiss (2001) was able to define two other foundational gates: the "de-repressor" gate and the "co-activator" gate. The main purpose of these gates is to perform intercellular communications using small molecules that freely diffuse across the cellular membrane (e.g. inducers). Both of these new gates have two inputs, namely a DNA-binding activator or repressor protein, and a small effector molecule that activates or inactivates the repressor. The co-activator gate has the same logic truth table as an AND gate, while the de-repressor has the truth table of "AND x (NOT y)", where y is the inducer small molecule.

Ron Weiss also demonstrated the fundamentals of "genetic process engineering" – taking existing genertic regulatory elements and modifying their DNA encoding so that they can be used in constructing complex *in vivo* digital-logic circuits. In particular, Weiss was able to mutate ribosome binding sites for the cI repressor and the operator for the bacteriophage lambda P(R) promoter so that the resulting cellular-logic gates had good noise margins and signal-restoration characteristics. This work is important because it shows how to synthesize biological components that can be combined to produce reliable circuits of significant complexity.

Engineered cell-to-cell communication One major thrust of this research is to make it possible to use living cells as a substrate for engineering, and to program colonies of simple bacterial cells to be test beds for the organizational principles of amorphous computing.

Knight and Weiss (described in Weiss, 2001) successfully cloned a set of Lux genes from *Vibrio fischeri* and *Photo-rhabdus luminescens.* This naturally occurring genetic circuit combines an intercellular cell density measurement with a complex biochemical light production enzyme cascade. They cloned this system, isolated the components with three distinct functions, and re-assembled those in several distinct ways.

The three sub-components are the autoinducer sender enzyme, responsible for creating the small signaling molecule N-acyl homoserine-lactone; the autoinducer response protein, responsible for controlled activation of transcription dependent on the concentration of

autoinducer; and finally the enzymatic light production cascade. The *P. luminescens* light production cascade was isolated specifically because of its ability to function at normal (37C) growth temperatures, unlike the corresponding version from *V. fischeri.* This approach of selectively isolating components from a variety of sources, with the explicit intention of creating an easily engineered set of system level components is one important project goal.

To perform this experiment, Weiss and Knight isolated a DNA fragment from *V. fischeri* which, when spliced into a plasmid, caused transformed *E. coli* colonies to glow. They then sequenced the complete region (a gene cluster with 8645 base pairs) and isolated from this structure the gene clusters responsible for light production, autoinducer production, and autoinducer response. By controlling expression of the autoinducer abd using light production as a sensing mechanism, they created a producer/sensor system that can transmit signals between cells.

Models of fundamental physics Erik Rauch (1999) developed amorphous discrete models of spacetime for fundamental physical systems. Discrete models are not unusual in physics: we investigate the foundations of physics with cellular automata, spin glasses, lattice gas models, and lattice gauge theory. All of these traditional models have a regular crystalline geometry and depend upon global synchronization. Mr. Rauch has shown that these constraints can be considerably relaxed to produce amorphous models (that are asynchronous and geometrically irregular) that exhibit realistic physical behavior: what Rauch has found is that to get physical behavior in an amorphous model one must constrain the interconnections of the lattice and the schedule of the updates to be "spacelike" and "timelike." However, it is not necessary for any one site to exploit detailed knowledge about what time other sites update, or about their precise spatioal arraygements.

In particular, Rauch has produced an amorphous model of the wave equation that exactly conserves energy and momentum. His amorphous model is not synchronous, nor does it depend upon locking schemes that effect synchronization in the asynchronous environment. In Rauch's scheme, the anisotropies due to the irregular interconnect average out in high-density systems.

Programmable materials Rather than build precisely engineered mechanical structures, one could program precise complicated structures starting from a single flexible mechanical base. Not only can one design many complex static structures from a single substrate, but one can also produce dynamic structures that can react to environment conditions or affect the environment. Such a *programmable material* would make possible a host of novel applications that blur the boundary between computation and the environment. Example applications may be a flexible car surface that can change structure exactly at the point of impact, a programmable assembly line that moves objects by producing ripples in specific directions, or manufacturing by programming global shapes on a single, flexible material.

Radhika Nagpal (2001) developed prototype model for controlling programmable materials. She showed how to organize a program to direct a sheet consisting of a vast number of autonomous, asynchronous, identically programmed, locally communicating, and irregularly placed agents, that can individually deform ("cells"), to cooperate to construct a large family of globally-specified predetermined shapes. She demonstrated this by presenting a language that allows a programmer to specify a sequence of folds, in a way inspired by Huzita's axioms for origami, that achieve the desired global arrangement. She showed how this language is compiled into a program that can be distributed to all of the agents. With a few differences of initial state (for example, agents on the edges of the sheet know that they are edge agents) the agents execute their copies of the program, interact with their neighbors, and fold up to make the desired shape.

Nagpal's techniques are quite robust. She has investigated and reported on the range of shapes that can be constructed using her method, and on their sensitivity to errors of communication, random cell death, and density of the cells. We believe that Nagpal's ideas will have an impact on the theoretical biology of differentiation and morphogenesis as well as on the development of technology for building more robust computer systems.

Large displays On the practical side, Rajeev Surati's 1997 Ph.D. thesis presents techniques for combining high-performance computing with feedback to enable the correction of imperfections in the alignment, optical system, and fabrication of very high-resolution display devices. The key idea relies on the measurement of relative alignment, rotation, optical distortion, and intensity gradients of an aggregated set of low-cost image display devices using a precision low cost reference. This idea provides a new technology for linearly scalable, bright, seamless, high-resolution large-scale self-calibrating displays (seamless video walls).

As proof of concept, Surati successfully constructed a prototype system for a self calibrating large-scale projection displays, such as seamless video walls. The system uses 4 projectors and 1 camera for calibration. The system creates the precision necessary to calibrate itself through computation and thus has no moving parts. This approach will make it possible to build large-scale projection displays at a vastly reduced cost, since it eliminates the need for precision construction and for calibration of the components.

Impact and applications

This research will have revolutionary impact by creating the foundational technology—the system architectures and algorithms—to amass and control computing agents at prices comparable to the raw material costs. The resulting computing systems, and the smart materials they engender, will be extremely robust in design. They will provide the possibility of bulk computation, through such media as smart paint, smart gels, and active building materials. Engineers will use these materials to reduce the need for strength and precision in mechanical and electrical apparatus through the application of computation. The resulting system architectures will be physically feasible at any scale.

They will be able to coordinate information from vast numbers of distributed sensors and to use it to control equally vast numbers of distributed effectors, thus enabling the construction of systems with unprecedented responsiveness to their environment.

This research also furnishes the implementation technology to harness living cells as a computational substrate. This will give us the power to design and construct biological cells and cell assemblies with prespecified behaviors. The ability of programmable cells to reproduce, to move, and to couple directly into the chemical environment, will have profound implications for materials manufacturing, sensing, effecting, and fabrication at the molecular scale. We are still at a primitive stage in the development of this technology, analogous to the early stages of the development of electronics at the beginning of the 20th century, and progress here will open a new frontier of engineering that could dominate the information technology of the next century.

The implications of this work for information technology are equally significant. Computer science is currently built on a foundation that largely assumes the existence of a perfect infrastructure. Integrated circuits are fabricated in clean-room environments, tested deterministically, and discarded if even a single defect is uncovered. Entire software systems fail with single-line errors. In contrast, biological systems rely on local computation, local communication, and local state, yet they exhibit tremendous resilience. No existing engineering framework creates such complex structure from simple, potentially faulty components or maintains such complex behavior of existing structures under dramatic environmental change. In order to meet this challenge, computer science will, over the next decade, undergo a transformation whereby its dominant organizational paradigms will be drawn from biology, rather than from physics and mathematics, as has been the case for the past half century.

The research we propose will help stimulate this transformation, with potentially enormous implications for both biology and computing. In biology, we can perhaps address longstanding questions of morphogenesis, and discover a computational understanding of the developmental process. In computing, we can perhaps learn to match nature's ability to create robust mechanisms out of unreliable components, thus realizing an ambition that has been with Computer Science since the dawn of the field.

Technology Transfer

This research project has a long time horizon. We do not expect extensive commercial or military application of these ideas very quickly, but we think that we will uncover principles that will become dominant in the next generation of advanced engineering systems. We have a history of cooperation with Hewlett-Packard Co., going back over fifteen years. This has included work both on the Scheme language and Scheme chips as well as collaboration on the application of mixed symbolic-numerical computing to controller design. We collaborated with HP on the design and construction of a Supercomputing Instrument Toolkit. For the amorphous computing project, we are currently discussing with HP Labs the possibility of collabrative research that would draw on HP Labs work

on the Teramac processor (reported in Science, June 1998).

Selected bibliography

References

[1] Harold Abelson, Don Allen, Daniel Coore, Chris Hanson, George Homsy, Thomas F. Knight, Jr., Radhika Nagpal, Erik Rauch, Gerald Jay Sussman, Ron Weiss, "Amorphous Computing, *Communications of the ACM*, Volume 43, Number 5, May 2001.

[2] Beal, Jacob. (2002) An Algorithm for Bootstrapping Communications. To be published in Proceedings of the International Conference on Complex Systems (ICCS) 2002.

[3] Beal, Jacob. (2002). Generating Communications Systems Through Shared Context. Master's Thesis, MIT, Department of Electrical Engineering and Computer Science.

[4] Daniel Coore, Radhika Nagpal and Ron Weiss, Paradigms for Structure in an Amorphous Computer, MIT AI Memo 1614, 1997.

[5] Daniel Coore, Botanical Computing: A Developmental Approach to Generating Interconnect Topologies on an Amorphous Computer, Coore, MIT PhD Thesis, 1999.

[6] Hearn, Robert A. "Building Grounded Abstractions For Artificial Intelligence Programming" Master's thesis, Massachusetts Institute of Technology, 2001.

[7] William J. Butera, Programming a Paintable Computer, PhD Thesis, MIT Media Lab, Feb. 2002

[8] Tom Knight and Gerald Jay Sussman, Cellular Gate Technology, First International Conference on Unconventional Models of Computation (UMC98), 1998.

[9] Mark Millonas and Eric Rauch,Trans-membrane signal transduction and biochemical Turing pattern formation¡/a¿, MIT AI Memo 1670, September 1999.

[10] Radhika Nagpal, Organizing a Global Coordinate System from Local Information on an Amorphous Computer, MIT AI Memo 1670, 1999

[11] R. Nagpal and D. Coore, An Algorithm for Group Formation in an Amorphous Computer, Intn'l Conf on Parallel and Distributed Computing Systems (PDCS'98), 1998

[12] Radhika Nagpal, "Programmable Self-Assembly: Constructing Global Shape using Biologically-inspired Local Interactions and Origami Mathematics", PhD Thesis, MIT, Department of Electrical Engineering and Computer Science, June 2001.

[13] Radhika Nagpal, "Self-Assembling Global Shape, using Ideas from Biology and Origami", to be published in Proceedings of the 3rd International Meeting of Origami Science, Mathematics and Technology (3OSME), March 2001.

[14] Nagpal, R. (July 2002) Programmable Self-Assembly using Biologically-Inspired Multiagent Control, First International Conference on Autonomous Agents and Multiagent Systems (AAMAS), Bologna, Italy.

[15] Nagpal, R. (June 2002) Programmable Pattern-Formation and Scale-Independence, International Conference on Complex Systems (ICCS), New Hampshire.

[16] Marcus A.M. de Aguiar, Hiroki Sayama, Erik M. Rauch, Yaneer Bar-Yam, and Michel Baranger. Stability and Instability of Polymorphic Populations and the Role of Multiple Breeding Seasons in Phase III of Wright's Shifting Balance Theory, Evolution, Phys. Rev. E 65: 031909 (2002).

[17] Erik M; Rauch, Discrete Amorphous Physical Models, MIT MS Thesis, 1999.

[18] Erik M. Rauch, Hiroki Sayama, and Yaneer Bar-Yam, The relationship between measures of fitness and time scale in evolution. Phys. Rev. Lett., 88: 228101 (2002).

[19] Gerald Jay Sussman and Jack Wisdom, with Meinhard Mayer *Structure and Interpretation of Classical Mechanics*, MIT Press, 2001.

[20] Ron Weiss, Thomas F. Knight, Jr., "Engineered Communications for Microbial Robotics," in DNA Computing: Sixth International Workshop on DNA-Based Computers, DNA 2000, Leiden, The Netherlands, June 2000. Available in Lecture Notes in Computer Science 2054, Springer, p. 1-16.

[21] R. Wiess, G. Homsy, T. Knight, Toward *in vivo* Digital Circuits DIMACS Workshop on Evolution as Computation, 1999.

[22] Werfel, J., M. Mitchell, and J. P. Crutchfield. Resource sharing and coevolution in evolving cellular automata. IEEE Transactions on Evolutionary Computation 4:388-393 (2000).

[23] Werfel, J. (2002). Implementing Universal Computation in an Evolutionary System. AI Memo AIM-2002-010.

[24] Ron Weiss, Cellular Computation and Communications using Engineered Genetic Regulatory Networks", PhD Thesis, MIT Artificial Intelligence Laboratory, 2001.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE 10/22/02

3. REPORT TYPE AND DATES COVERED Final 9/1/96 - 6/30/02

4. TITLE AND SUBTITLE
Amorphous Computing

5. FUNDING NUMBERS
N00014-96-1-1228
G

6. AUTHOR(S)
Hal Abelson, Gerald Jay Sussman, and Thomas F. Knight

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Mass Institute of Technology
77 Mass Ave
Cambridge MA 02139

8. PERFORMING ORGANIZATION REPORT NUMBER
1? (Is this the 1st report?)

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research
Code 252, Diane Bales
Ballston Towers One
800 North Quincy St.
Arlington, VA 22217-566

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
—

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release

12b. DISTRIBUTION CODE
—

13. ABSTRACT *(Maximum 200 words)*

The objective of this research is to create the architectural, algorithmic, and technological foundations for exploiting programmable materials. These are materials that incorporate vast numbers of programmable elements that react to each other and to their environment. Such materials can be fabricated economically, provided that the computing elements are amassed in bulk without arranging for precision interconnect and testing. In order to exploit programmable materials we must identify engineering principles for organizing and instructing myriad programmable entities to cooperate to robustly achieve pre-established goals, even though the individual entities are unreliable and interconnected in unknown, irregular, and time-varying ways.

14. SUBJECT TERMS
Insert keywords identifying major subjects in the

15. NUMBER OF PAGES

16. PRICE CODE
—

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
NA	NA	NA	UL